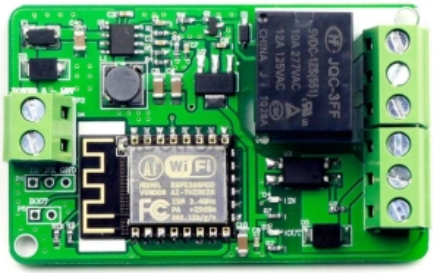
µC eXperiment
Experimentation with microcontrollers.

Blynking an IoT Yunshan ESP8266 250V 10A AC/DC WIFI Network Relay Module

Posted on December 18, 2016



I purchased a few of these Yunshan Wifi Relays through ebay for approximately $7.50US. The device should be perfect for use in simple IOT projects which require controlling household AC power. The onboard JQC-3FF relay is rated to 250VAC or 30VDC at up to 12A. There are routered slots between the high voltage PCB traces for circuit isolation and arc-over protection. Transient voltage suppression is incorporated on both the board power supply and the photocoupler (see description below) input line.

The device requires a power supply between 7 and 30V DC. I unsuccessfully attempted to run it with an inexpensive 5V, 2A wall-wort, even though the onboard MP2303 buck converter is rated down to 4.8V. I did get it to operate successfully using a 9VDC wall-wort.

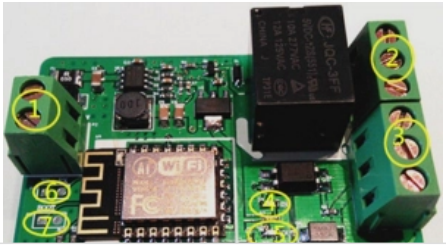
The device contains an integrated ESP8266-12E, but appears to only use the GPIO4 and GPIO5 pins. That was a disheartening discovery because it discards a significant amount of functionality inside the ESP8266 WIFI module. Hence the ESP8266 low power, wake from sleep provisions (where GPIO16 and RESET need to be linked together) would require some skillful soldering of the module's exposed pins.

The good news is, programming the module is very easy, as I discuss later. I also found the overall build quality of my device to be above the typical level found on ebay-sourced Chinese electronics.

The ebay listing contained a link to a zip file, entitled *U4648-datasheet*, which contained example programs, schematics, and a Chinese manual. Through the Google translation service I managed to translate the manual, but there's no reason to do that, as there isn't much there. More information can be learned from a quick study of the schematic and the board itself.

Module Description

The Chinese manual presents the following limited module description:



3 – IO input terminal.
4 – Enter the status indicator, IO input high when lit, blue light.
5 – The relay output status indicator, the relay is turned on, the red light.
6 – TTL serial output.
7 – Boot mode selection jumper.

Board Connectors

Here are the connections on my board:

A: 7-30V+ DC power supply
B: Power supply ground
C: Normally closed (NC) relay contact
D: Common (COM) relay contact
E: Normally open (NO) relay contact
F: 5V+ out
G: ESP8266 GPIO5 Optocoupler Input
H: Ground (isolated optocoupler input)

AP MODE Webpage

I was easily able to connect a 9V power supply to the A-B connector (see above picture and connector description) and control the device via WIFI. To do this, simply connect your computer or phone to the *yunshan_wifi_xx_xx_xx* network (where it appears the xx are hexadecimal numbers pulled from the ESP8266 MAC address). My device responded to the supplied password of *yunshan123456789*. Once a connection was established, I simply entered the IP address of 192.168.4.1 into my browser. Once there, I was greeted by a Chinese web page, the translation of which appears below. From this webpage, I was able to open and close the relay. The status of the GPIO5 optocoupler input is also displayed on this webpage.

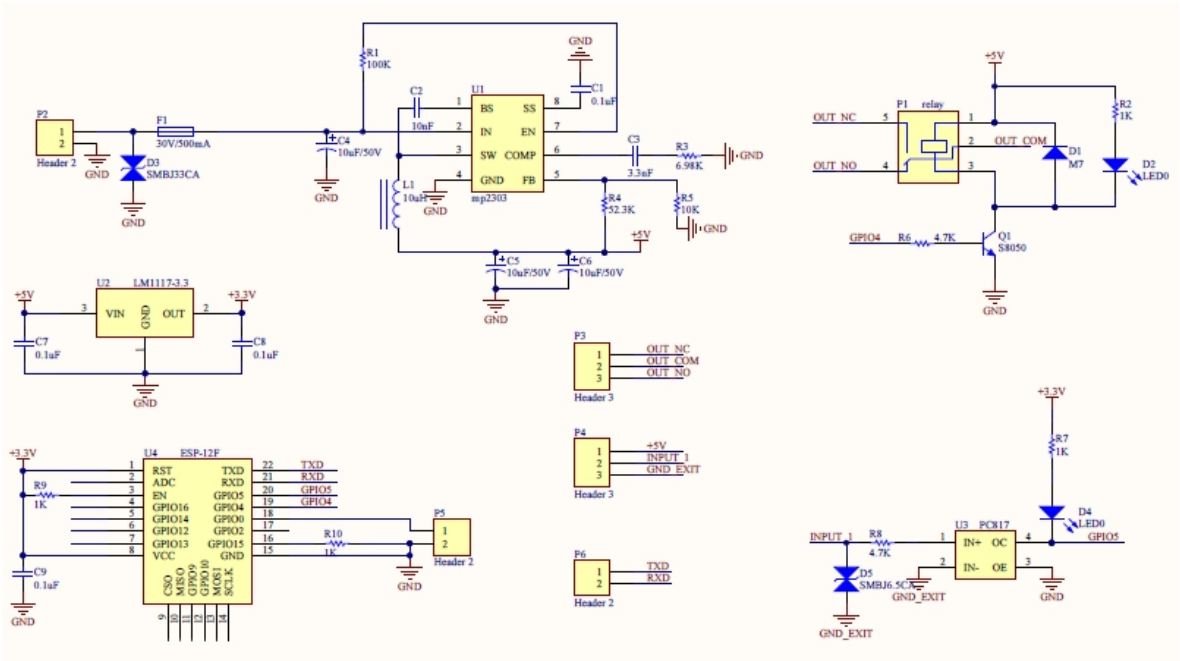
Since I have big IOT home automation plans for these devices, my next task was to attempt a re-program of the onboard ESP8266 module. For a quick test, I uploaded the traditional Arduino IDE ESP8266 blink program, and

On the lower left portion of the PCB is a section that grants access to the ESP8266 pins for programming (see the above photo). These same pins are also useful for TTL serial output purposes (debugging, etc.). Separate 2 and 3-pin headers will need to be soldered into these connector holes (labeled P5 and P6). The ESP8266 GPIO4 controls the relay through a 2N3904 transistor. Setting GPIO4 high, causes the relay to close the NO contact with Common and the NC contact to open. Additionally, taking connector "G" high causes GPIO5 to also go low isolated via a PC817 photocoupler. On my board the blue LED is connected to GPIO2, and can be illuminated by pulling the pin low.

To program the ESP8266 module, I connected the TX, RX and ground pins of connector P6 to a SparkFun USB FTDI programmer, and jumped the two pins of connector P5 together when I was ready to upload. Connector P5 grounds GPIO0 and GPIO15, sending the device into bootloader mode. If you have trouble programming the ESP8266 like I did on the first attempt, ensure you also ground your FTDI device through the P6 connector.

A very good introduction to the ESP8266 module can be found here. Excellent programming information for the individual ESP8266 modules is also widely available (two examples: ESP8266-01 and ESP8266-12e).

Board Schematic



Blynk Relay Control Application

```
/*************************************************************************
 * Title: Simple ESP-8266 blynk/yunshan wifi relay control
 * File: esp8266_yunshan_relay.ino
 * Author: James Eli
 * Date: 12/25/2016
 *
 * This program controls a Yunshan wifi relay module communicating through
 * the onboard esp-8266-12e module. The module is controlled from the
 * internet via the Blynk cloud app.
 *
 * Notes:
 *  (1) Requires the following arduino libraries:
 *      ESP8266
 *      Blynk
```

```
 ************************************************************************
 * Change Log:
 *   12/25/2016: Initial release. JME
 *   12/31/2016: Added input pin status. JME
 *   01/15/2017: Added volatile. JME
 ************************************************************************/
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// Esp8266 pins.
#define ESP8266_GPIO2    2 // Blue LED.
#define ESP8266_GPIO4    4 // Relay control.
#define ESP8266_GPIO5    5 // Optocoupler input.
#define LED_PIN          ESP8266_GPIO2
// Blynk app authentication code.
char auth[] = "***";
// Wifi SSID.
const char ssid[] = "***";
// Wifi password.
const char password[] = "***";
// Flag for sync on re-connection.
bool isFirstConnect = true;
volatile int relayState = LOW;    // Blynk app pushbutton status.
volatile int inputState = LOW;    // Input pin state.

void setup() {
  pinMode( ESP8266_GPIO4, OUTPUT );       // Relay control pin.
  pinMode( ESP8266_GPIO5, INPUT_PULLUP ); // Input pin.
  pinMode( LED_PIN, OUTPUT );             // ESP8266 module blue LED.
  digitalWrite( LED_PIN, LOW );           // Turn on LED.
  Blynk.begin( auth, ssid, password );    // Initiate Blynk conection.
  digitalWrite( LED_PIN, HIGH );          // Turn off LED.
}

// This function runs every time Blynk connection is established.
BLYNK_CONNECTED() {
  if ( isFirstConnect ) {
    Blynk.syncAll();
    isFirstConnect = false;
  }
}

// Sync input LED.
BLYNK_READ( V2 ) {
  CheckInput();
}

// Blynk app relay command.
BLYNK_WRITE( V0 ) {
  if ( param.asInt() != relayState ) {
    relayState = !relayState;                  // Toggle state.
    digitalWrite( ESP8266_GPIO4, relayState ); // Relay control pin.
    Blynk.virtualWrite( V1, relayState*255 );  // Set Blynk app LED.
  }
}

// Debounce input pin.
int DebouncePin( void ) {
  // Read input pin.
  if ( digitalRead( ESP8266_GPIO5 ) == HIGH ) {
    // Debounce input.
    delay( 25 );
    if ( digitalRead( ESP8266_GPIO5 ) == HIGH )
      return HIGH;
  }
  return LOW;
```

```
    Blynk.virtualWrite( V2, inputState*255 );
    inputState = !inputState;
  }
}

// Main program loop.
void loop() {
  Blynk.run();
  CheckInput();
  //yield(); //Updated: 3/8/2017
}
```

TCP Client Demo

Here is a basic server which responds to TCP client HTTP GET commands (added 1/8/17):

```
#include <ESP8266WiFi.h>

// Esp8266 pinouts
#define ESP8266_GPIO2    2  // Blue LED.
#define ESP8266_GPIO4    4  // Relay control.
#define ESP8266_GPIO5    5  // Optocoupler input.
#define LED_PIN          ESP8266_GPIO2
// WiFi Definitions.
const char ssid[] = "***";
const char pswd[] = "***";
WiFiServer server( 80 );
volatile int relayState = 0;      // Relay state.

void setup() {
  initHardware();
  connectWiFi();
  server.begin();
}

void GetClient( WiFiClient client ) {
  // Read the first line of the request.
  String req = client.readStringUntil( '\r' );
  Serial.println( req );
  client.flush();

  String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE HTML>\r\n<

  if ( req.indexOf( "OPTIONS" ) != -1 ) {
    s += "Allows: GET, OPTIONS";

  } else if ( req.indexOf( "GET" ) != -1 ) {
    if ( req.indexOf( "open" ) != -1 ) {
      // relay on!
      s += "relay on!";
      relayState = 1;
      digitalWrite( ESP8266_GPIO4, 1 ); // Relay control pin.

    } else if ( req.indexOf( "close" ) != -1 ) {
      // relay off!
      s += "relay off!";
      relayState = 0;
      digitalWrite( ESP8266_GPIO4, 0 ); // Relay control pin.

    } else if ( req.indexOf( "relay" ) != -1 ) {
      if ( relayState == 0 )
        // relay off!
        s += "relay off!";
      else
        // relay on!
```

```
      else
        s += "input io is:1!";

    } else if ( req.indexOf( "MAC" ) != -1 ) {
      uint8_t mac[WL_MAC_ADDR_LENGTH];
      WiFi.softAPmacAddress( mac );
      String macID = String( mac[WL_MAC_ADDR_LENGTH - 5], HEX) + String( mac[WL_MAC_
                     String( mac[WL_MAC_ADDR_LENGTH - 3], HEX) + String( mac[WL_MAC_
                     String( mac[WL_MAC_ADDR_LENGTH - 1], HEX) + String( mac[WL_MAC_
      macID.toUpperCase();
      s += "MAC address: " + macID;

    } else
      s += "Invalid Request.<br> Try: open/close/relay/io/MAC";

  } else
    s = "HTTP/1.1 501 Not Implemented\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE HT

  client.flush();
  s += "</html>\n";

  // Send the response to the client.
  client.print( s );
  delay( 1 );
  Serial.println( "Client response sent." );
}

void loop() {
  // Check if a client has connected.
  WiFiClient client = server.available();
  if ( client )
    GetClient( client );
}

void connectWiFi() {
  byte ledStatus = LOW;
  Serial.println();
  Serial.println( "Connecting to: " + String( ssid ) );
  // Set WiFi mode to station (as opposed to AP or AP_STA).
  WiFi.mode( WIFI_STA );

  // WiFI.begin([ssid], [passkey]) initiates a WiFI connection.
  // to the stated [ssid], using the [passkey] as a WPA, WPA2, or WEP passphrase.
  WiFi.begin( ssid, pswd );

  while ( WiFi.status() != WL_CONNECTED ) {
    // Blink the LED.
    digitalWrite( LED_PIN, ledStatus ); // Write LED high/low.
    ledStatus = ( ledStatus == HIGH ) ? LOW : HIGH;
    delay( 100 );
  }

  Serial.println( "WiFi connected" );
  Serial.println( "IP address: " );
  Serial.println( WiFi.localIP() );
}

void initHardware() {
  Serial.begin( 9600 );
  pinMode( ESP8266_GPIO4, OUTPUT );       // Relay control pin.
  pinMode( ESP8266_GPIO5, INPUT_PULLUP ); // Input pin.
  pinMode( LED_PIN, OUTPUT );             // ESP8266 module blue LED.
  digitalWrite( ESP8266_GPIO4, 0 );       // Set relay control pin low.
}
```

Share this:

Twitter Facebook 8 Print Reddit

Like

2 bloggers like this.

Related

Using a Sonoff S31 with Blynk
In "arduino"

Controlling an IoT Yunshan ESP8266 WIFI Network Relay via an Amazon Echo/Alexa
In "iot"

Re-programming a Lyasi ESP-8285 WiFi Wall Switch
In "arduino"

About Jim Eli
µC experimenter
View all posts by Jim Eli →

This entry was posted in iot and tagged esp8266, iot. Bookmark the permalink.

114 Responses to *Blynking an IoT Yunshan ESP8266 250V 10A AC/DC WIFI Network Relay Module*

Peter *says:*
December 30, 2016 at 5:15 am

Hi Jim,
her article gives me hope. Try now for several days to find a solution to reprogram this circuit.
First of all, all circuits that I bought were broken by the hardware. All the coils had been soldered incorrectly and the fuses were also broken.
Repaired, I can now AT commands and adjust also. But I can not take a separate function from any of the circuits.
The circuit can not be reprogrammed.
My question to you:
You set the P5 (jumper) permanently during programming and RX TX = 1: 1 or RX cross TX.?
I have tried various 3.3 V FTDI's some work not at all (seemingly felled cn).
Do you have a solution for me?
thank you
Peter

Sorry 😟 Google Translate

December 30, 2016 at 10:06 am

Peter,
So sorry you are having these problems. My devices worked correctly from the beginning. I hope I can be of some assistance.
(1) You must connect a separate and appropriate power supply independent of the FTDI programmer when programming. The FTDI programmer probably has insufficient power.
(2) P5 connector is meant to be jumped together only during programming. When programming is complete, remove the jumper and reboot.
(3) Connect your FTDI RX to P6 TX, and FTDI TX to P6 RX, and FTDI ground to the ground.
(4) Often when I attempt to program, the download will not occur. In this case, I simply remove the power from the device and reboot with the jumper in place, and then re-attempt the download. There is no need to disconnect the FTDI programmer. This usually works.
(5) I program via the Arduino IDE. Make sure you are selecting a generic ESP8266 module.
(6) Try programming at lower baud rates before trying a higher rate.
Jim

Reply

Herman Eggink *says:*
January 1, 2017 at 11:28 am

Hi Peter, same with me: fuses appear to be broken and coils rotated 90 degrees. weird…

Reply

Jim Eli *says:*
January 1, 2017 at 12:01 pm

The devices I received have a single 5-pin JQC-3FF relay. The pins are asymmetrical, so nearly impossible to solder wrong (but I guess its possible). Also my boards have just a single fuse on the 30VDC board power input line. Is it possible someone mistakenly applied AC power here?

Reply

Peter *says:*
January 13, 2017 at 11:25 am

Hello,
Sorry Jim a little addendum:
It's been told wrong. It is the coil of the voltage regulator it is incorrectly soldered, it creates a short circuit and the fuse 500mA is destroyed.
Three unsung boards had the fuse already broken.
There was no alternating voltage, I'm an electronics engineer
My board is working, my problem was the reset. The board may only be reset shortly before the end of the slow compilation? (Timeout)
thank you
Peter

Herman Eggink *says:*
January 22, 2017 at 12:40 pm

Hi Peter, can you show me a before/after picture?

Reply

Kenanda *says:*
January 1, 2017 at 3:48 pm

Just a question.... this yunshan device is supposed to receive tcp commands. The question is... , what command do I use to turn the relay on and off with my tcp client? I did not find it.

Reply

Jim Eli *says:*
January 1, 2017 at 4:07 pm

I would suggest looking at the TCP Client Demo source code in the download link.

Reply

sun energy egypt *says:*
January 11, 2017 at 2:53 pm

what is the command do I use to turn the relay on and off with my tcp client? I did not find it.

Reply

Jim Eli *says:*
January 11, 2017 at 4:43 pm

RTSL. “open relay!” and “close relay!”

Reply

Ivan Vachovsky *says:*
January 30, 2017 at 2:31 pm

Great ! Those are working. Thank You Jim.
Can you also tell us please the commands for
1. Input status
2. Relay status only (without activating or deactivating the relay)
3. Other commands if any.

Jim Eli *says:*
January 30, 2017 at 3:43 pm

Try “get relay state!”, “get io state!”, and “get MAC!”.

Ivan Vachovsky *says:*
January 30, 2017 at 4:04 pm

Thanks Jim. All of the commands you suggested did work.

Ferenc Tolnai *says:*
January 18, 2017 at 3:37 pm

hi guys, I purchased one of these from china.
I can only get start if i bypass the fuse F1, by connecting the + to the right side of the diode (M7) should this not be a fuse instead as?

It turned out that the M7 diode was soldered in in wrong direction. I turned it around 🙂

Reply

Ivan Vachovsky *says:*
January 30, 2017 at 10:28 am

Yes. I had this M7 diode experience as well with unit purchased from Amazon. Furthermore, this unit came blank (not programmed) and I returned it. Over 20 different vendors are selling same device on eBay. Mostly from China and 1-2 local in the U.S. Bought another one from US Vendor on eBay. I managed to activate the web interface for this one, but FTP doesn't work. Port 8000 is open and listening but commands like "relay on!" or "open relay!" do not work.

Attila Tajti *says:*
December 28, 2017 at 6:27 am

I may have the same issue…how can you tell which is the correct direction of the diode?

Ivan Vachovsky *says:*
January 23, 2017 at 10:09 pm

Jim,
Like you I had big plans for this device and purchased 5 from Amazon along with 5 12V DC power supplies..
Unlike you I'm not set to do the programming of it.
I trusted some Amazon reviews that the device is not perfect but workable and can be accessed by HTML and FTP similar to what you describe.
To my great disappointment the device creates only own unsecured WiFi called AI-THINKER. It has IP address 192.168.4.1 and does respond to pings. That's all. I scanned for open ports/services – none. I specifically looked for TCP service on port 8000 (link from Amazon review mentioned this port specifically for FTP) – none. No web service either. It seems that this is blank AI-THINKER device not programmed with Yunshan features as described at Amazon.
Based on your experience, am I missing something here or should I return everything to Amazon.
Did you get yours from US vendor on eBay or from China?

Reply

Jim Eli *says:*
January 24, 2017 at 7:47 am

Ivan,
I didn't spend much time playing with the installed software, other than to test the AP mode webserver. I briefly looked at the demo software I mention in the download zip file (see link in blog). Response for a TCP client appears to be implemented in an entirely different program than the version that was installed on my devices. My guess is that what you are looking for isn't there. I suggest you install what you need, as programming is really easy (especially when using the Arduino IDE). Good luck.
Jim

Reply

Jason N *says:*
December 3, 2017 at 4:33 pm

I got the same AI_THINKER. I can ping as well, but port scanner shows no port open

Reply

Ivan Vachovsky *says:*

Thanks for the response. Appears that Amazon is selling visually the same device with different programming (blank actually) while the description is for Yunshan programmed device.
Anyhow, I returned it to Amazon and bought true Yunshan device from eBay.
It matches your description above perfectly all the way down to your "Program Upload" paragraph.
Since I have no intention of uploading programs (not qualified for that) I was wondering if you can help me setting up the device. I can connect to it trough its own wifi network to address 192.168.4.1. I can see the web page, however none of the buttons work. Browser tells me "No Internet connection". Of course no Internet since I'm connected to the device alone. I also managed connecting to TCP port 8,000 however it will not respond to any of the commands "relay on!" or off. So basically I'm stuck again.

Thanks,

Ivan

Reply

Ivan Vachovsky *says:*
January 27, 2017 at 10:43 am

P.S. Managed to connect the device to my WiFi network. I don't know why but it needed Internet connection for that. So I used a desktop computer with wired connection to the Internet and WiFi connection to Yunshan WiFi.
Now I can control the relay over the web interface. However the TCP commands "relay on!" and "relay off!" do not work.
My postings seem somewhat off topic I just couldn't find a better place to share my experience and help others who may struggle with the same Chinese problems.
My objective is to integrate the relay in my IoT system by standard means (HTTP or FTP control) There are many other solutions on the market however most of them are proprietary with secret hidden protocols and hard to integrate.

Reply

Nikola Stojanovic *says:*
March 13, 2017 at 6:30 am

Hi Ivan, I wasn't able to connect it to my WiFi network. I've put the network ID and password in Yunshan WiFi settings over 192.168.4.1 web interface, using my laptop. If I try to connect to web interface over my phone, the commands doesn't work, because no internet is available. Is there a catch?

Reply

Ivan Vachovsky *says:*
February 1, 2017 at 12:37 pm

Hi Jim,

It's me again. As the French proverb goes "Appetite comes with the eating". I have the Yunshan relay completely under control now incl. commands and responses. Thank you again for the help on the commands.
I'm now interested in extending the functionality as explained below. Is it possible for you to write the code that is needed and guide me step by step of how to upload it. Contact me privately how to get paid for the time and effort.
In many instances one needs to introduce "timeout" feature when controlling significant power trough home automation – examples – pool pumps, sauna, steam room, electric room heaters etc. It can be implemented trough software but it defeats the purpose since the timeout is there to kick-in when the controlling software/system fails. That's why it is preferably that the timeout is implemented outside of the control system and is build-in in the firmware/hardware at lower level.
Long story short I need the relay to execute a command like this

relayon, t

where t= timeout in seconds,

In addition to timeouts, this function will come handy in creating short pulses that are necessary to operate some devices such as garage doors for example.

of course such function can be achieved by adding external timer relays, but it wouldn't be an elegant solution.

Reply

Tiju *says:*
February 2, 2017 at 6:10 am

Thanks for the code! got me running real fast 🙂
Only using the Relay for output atm, how would I use the input?
do I need to ground it or apply 5V or more?
Would like to connect it to a Speaker for a wireless doorbell and get notified when it rings 🙂
Cheers!

Reply

Zeric E *says:*
February 4, 2017 at 10:19 pm

Ivan, I have a suggestion for you if you haven't already worked out another solution. There is firmware for the esp8266 called Esp Easy that has considerable flexibility to perform different tasks in different esp8266 based hardware without requiring re-compiling or reloading it into the device. It can be downloaded in binary form so you don't need to setup the Arduino environment to compile or load it into a device. There is a simple windows tool to upload the firmware to the esp8266. I have successfully loaded it into this relay board as well as other esp8266 based devices and been able to do what I needed to do with my home automation system.

Even though I have a working Arduino IDE that I've used to do things with the esp8266, my go to solution is Esp Easy for these simple esp8266 IOT devices.

For the USB/TTL hardware interface, I prefer ch340g based converters even though I own FTDI and CP2102 ones as well. The ch340g boards are cheap and I've never had any issues with them (they also work fine with the Arduino IDE). Whatever interface you use, be sure it can work with 3.3v logic levels, most boards have a switch or jumper to go between 3.3v and 5v. If you have an Arduino UNO, that should also work to do the programming but it's not as straight forward as a stand alone USB/TTL converter and I haven't tested it with the uploader.

If this is a path you want to explore, I'll be happy to give you some pointers.

Reply

billskeen *says:*
July 24, 2017 at 4:18 pm

Thanks for the heads up about ESP Easy. It was fairly easy to upload firmware (after getting a flash program that would work. flashESP8266.exe did not work for me on Windows 10, returning a file not found error).

After simple config to connect to my network, I can now use simple http get cmd to turn relay on and off.

Reply

Ivan Vachovsky *says:*
February 5, 2017 at 3:04 pm

Zeric, Thanks a lot for the proposal. I did hire a freelancer to do the programming for me but I'm also curious and eager to try Esp Easy for other projects possibly. I will appreciate a guidance of how to do it. I have this device for uploads – Qunqi 3.3V

February 6, 2017 at 6:30 am

Ivan, Your Qunqi adapter should be fine as long the device driver is working okay with your PC. Most people on Amazon didn't seem to have any issue with it. Just be sure you have the jumper set to 3.3v. There is quite a bit of information on Esp Easy here: http://www.letscontrolit.com/wiki/index.php/ESPEasy

Esp Easy was originally designed for sensors, but it works well controlling simple on/off things like relays too. Enabling "rules" in the Tools > Advanced menu enables it to perform some simple definable autonomous function (ex. when this pin changes state, turn on this other pin for XX seconds). I use it with an MQTT broker, but other types of access are also possible such as HTTP. I've been impressed with the flexibility without having to re-compile/upload.

It's worth taking a look at when you have a chance.

Reply

Ivan Vachovsky *says:*
February 7, 2017 at 4:07 pm

Thanks will do.

Jason N *says:*
November 21, 2017 at 8:33 pm

Ivan,

I'm looking for the same functionality as you. I first want it to be able to take an http command to momentary close the relay for my garage door. Did your freelancer implement this, and if so, are you willing to share? I'd contribute to the cost

Jason

Reply

Larks *says:*
February 7, 2017 at 12:44 am

Hi jim, in your opinion is it possibile use the Optocoupler Input for a 220v Main sense?

Reply

Jim Eli *says:*
February 7, 2017 at 10:30 am

Larks,
My version of the device came with a PC817 Opto-Coupler. I think it's only spec'd to handle around 30 volts. Since 5V out is shared on its connector, I'm guessing it was designed to be used at that level.

Reply

olivier *says:*
February 7, 2017 at 11:36 am

If we use the bridge R1 and D1 as on the following assembly. It should work?
http://www.electroschematics.com/6994/220v-power-line-interface/

Reply

This will not burn anything but is not a good solution because of the sine wave there will be periods of low current and false readings of 1 when power is present. First determine what current I holds solid steady 0. Say it is 1 ma. Than build a simple rectifier one diode (better 4 diodes bridge widely available) and one capacitor will do the job for such low current. Than measure the voltage on your rectifier say 200 V. Connect the IO input trough a resistor R. R=U/I. For the example R = 200/0.001 = 200Ko. Check the power on the resistor (approximately ignoring the voltage drop on the optocoupler) P=U*I. For the example P=200*0.001= 0.2 W. This is acceptable power level, however if the real values show power over 1.0 W it may get too hot.

Reply

Larks *says:*
February 9, 2017 at 6:02 am

But the input has just an R of 4,7k on input in the schema… In this case is sufficient a Diode before? Which task has the parallel diode(D5)?
Thanks for info, i'm newbie 🙂

Reply

Ivan Vachovsky *says:*
February 9, 2017 at 11:41 am

The parallel diode prevents negative portion of the sine wave from hitting the opto-coupler. This schema will work if implemented completely as shown. It will not work if you implement only the portion that feeds the opto-coupler and leave the rest to Yunshan board. Follow my instructions above for a much simpler solution for the Yunshan board.

Reply

Keith *says:*
February 18, 2017 at 10:30 am

So what all applications does this opto isolator have? is it for just input or output too?

Reply

Nick P *says:*
February 27, 2017 at 1:13 am

Cool little project Jim, thanks for leaving it for people to try.
I've successfully burnt your code to my Yunshan board and made a small Blynk app to control it. The only thing is that it loses the wifi connection after a period of time and doesn't seem to attempt to re-connect so I just get a message from Blynk telling me that "device went offline at xxxx" and I have to reboot the board to get everything working again. Any ideas?

Reply

Jim Eli *says:*
February 27, 2017 at 6:47 am

Nick,

My device has dropped offline a few times too. I haven't had the time to fully investigate this, and I'm open to suggestions. Here are a few of my initial thoughts: (1) Is the power supply adequate? (2) Is it an issue with Blynk, ESP8266 or the WIFI router side? (3) Would the the ESP8266 boot ROM log shed any light?

However, since some of my other devices don't have this problem, I'm thinking it might be a HW/PS issue.

Nick P *says:*
February 27, 2017 at 7:23 am

Hi Jim,
The power supply is fine (it's a regulated lab power supply, easily big enough). The router to ESP Wifi connection is strong and so I doubt that's a problem. Yesterday it was online for some hours and then went off shortly after midnight. This morning I tried to reboot the unit a few times without success (just blue LED showing no connection). I tried re-flashing the code to the ESP and that worked.
The only thing I can think from my end is that our connection to the internet can drop out for a minute or two (this often happens around the time the unit went offline, just after midnight) so hence my earlier question. I'm afraid code-wise I'm a newbie at this stuff so can't really help de-bug at that level. I've tried a couple of Sonoff devices in the same place and they work fine.
Wonder if anyone else has had issues like this?
Thanks!

Reply

Jim Eli *says:*
February 27, 2017 at 7:33 am

What voltage are you driving the device at?

Reply

Jim Eli *says:*
February 27, 2017 at 7:50 am

Just looked at the code, and here's a thought, try eliminating the "yield" command inside the main Loop.

Nick P *says:*
February 27, 2017 at 9:53 am

It's 12V.
Will try the yield thing and let you know. Thanks

Reply

Nick P *says:*
February 27, 2017 at 10:02 am

Well, code is still working after removing the yield. Will leave the module on and unattended and let you know. In fact I might try turning the wifi network off for a minute and see if the module tries to re-connect by itself.

Reply

Nick P *says:*
February 27, 2017 at 10:07 am

Just tried turning off the wifi and the module re-connects and starts working again when the wifi comes back up.

Reply

Jim Eli *says:*
February 27, 2017 at 10:20 am

Nick P *says:*
February 28, 2017 at 11:40 am

Hi Jim,
Well, that seems to have sorted it. The module was on for over 24hrs without a problem. Whilst doing other stuff nearby I knocked the power supply leads a couple of times and the power was lost briefly; both times the module powered back up and connected to wifi and the Blynk server within 3-4 seconds. I'll keep you posted if anything changes. Thanks!

Reply

Bart *says:*
March 2, 2017 at 11:39 am

Hi,

Did you really manage to make the relay switch? I've bought three of those network boards but no luck switching the relay. After flipping a diode where a fuse needs to be, I can make everything work and when I switch the relay the output LED will go on but the relay does not switch. After removing the MP2303 Buck converter on one of them and connecting 5V directly to the LM1117-3.3V it does switch the relay. So I checked the output of the MP2303 on the other two boards and it seems to be configured to give 3V3. So it provides 3V3 to the LM1117-3V3 which gives 2V5 on his output, enough to power the ESP8266 but the 3V3 is to little to switch the relay…
I checked the values on the boards and they are all the same as on your schematic. It's just weird I'm having this issue on three boards?

Reply

Bart *says:*
March 3, 2017 at 12:16 am

After reading Peter his comment, I can confirm that the coil of the voltage regulator was placed incorrect. Although there are no fuses but diodes on the boards, replacing the coil solved the issues.
Conclusion: bad build quality although different suppliers..

Reply

Bart *says:*
March 2, 2017 at 11:46 am

BTW, I think when you replace the diode with a real fuse you will not have a voltage drop of 0.7V over the diode. Giving you the possibility to use a 5V power supply (5-0.7 = 4.3V which is below the 4.8V minimum)

Reply

Ferenc Tolnai *says:*
March 2, 2017 at 1:00 pm

now I got hold of another one, this has Resistor R8, 52kOhm instead of 4.7k, so the input doesnt work, what a quality…. how difficult would it be to test 2 out of the 2 functions.

Reply

Nick Procktor *says:*
March 2, 2017 at 2:09 pm

It seems that there's a large variation in the build quality of these boards. I bought mine (just one) from Ebay and it worked first time. The link is here…

Nick Procktor *says:*
March 2, 2017 at 2:13 pm

Also, for the optocoupler input you need to provide a second ground reference. This input is totally isolated and needs it's own ground and voltage supply to trigger the opto.

Reply

Timo *says:*
March 13, 2017 at 5:10 am

thank you very useful information – mine had has well diode soldered wrong around.

Reply

Nikola Stojanovic *says:*
March 14, 2017 at 4:04 am

Hi, I intended to hook up the device to my home network and access it with TeamViewer. There is a problem with hooking up. I've managed to access the web interface via 192.168.4.1 and I've entered user name and password of my home wireless network. However, I'm only able to change the state of relay if I'm directly connected to its wireless network (yunshan_wifi_xx_xx_xx) via laptop. This means that I cannot change its state from my phone, since the phone disconnects from home wireless network to connect with relay wireless network. It seems that relay is not connected to home wireless network at all.
Did anyone have similar problem?

Reply

Nick P *says:*
March 14, 2017 at 4:47 am

If you can program the board using an RS232 converter and PC then upload the code at the top of this page. You then use a program on your phone called Blynk and use that to control the module. The software that comes already installed when you buy the board is not very good and just a demo really, in my opinion.

Reply

Nikola Stojanovic *says:*
March 16, 2017 at 10:13 am

Thanks, Nick. That's too complicated for me, though. I'm just a guy trying to find a way to make it work. 🙂 Have you tried to change the state of the relay over the web with its pre-installed software, as described in Chinese manual?

Reply

Michael Barkley *says:*
April 4, 2017 at 8:22 pm

Since my board, does not seem to produce an SSID, when it's powered up, am I correct in thinking it's not properly programmed?

Reply

Jim Eli *says:*
April 9, 2017 at 5:22 am

Make sure you provide enough voltage to the board, the esp8266 needs an appropriate amount of juice before it

Michael Barkley *says:*
April 9, 2017 at 8:01 am

Thanks, I have been merely using a 9 volt battery, will try a 12 volt source instead.

Stefan *says:*
April 18, 2017 at 4:36 am

Hi Jim,
did you managed to read the input correctly? For me, reading GPIO5 is always “1”, even if I connect “F” to “G”.
I flashed nodeMCU to my board, anything else is running fine, also switching the relay. Putting GPIO5 to OUTPUT-State is also possible, the little blue LED can be put on and off.

Anybody else some experience with reading input?

Reply

Jim Eli *says:*
April 18, 2017 at 6:01 am

The optocoupler has a (properly) isolated ground, so the 5V out is reference a different ground. Therefore, simply connecting F to G will do nothing.

Reply

Stefan *says:*
April 19, 2017 at 1:48 pm

Thank you Jim,
that solved my problem. Connecting the external ground (“B”) to connector “H” did the job.

Marco *says:*
April 25, 2017 at 1:51 am

HI thanks for sharing your knowledge with thit esp8266 board

I use this with http://souliss.net/
The opto isolated inot don t work proply.
I think is a hardware issue

I read r8 is only 1,5 kohm not 4.7

When I power the board the input is 0 when i change value the input became 1
the led power up, but when I try to return to lower state the input remains 1 and the led Is always on

Reply

Marco *says:*
April 26, 2017 at 12:23 am

Ok I shortcut the input led and now work! I think the led is mounted reversed

Reply

---

roberto carlos *says:*

can you help me?
I can not make it work with arduino IDE … help
http://www.chinalctech.com/index.php?_m=mod_product&_a=view&p_id=1204
http://www.chinalctech.com/index.php?_m=mod_product&_a=view&p_id=1205

Reply

Sukhbir Singh *says:*
July 24, 2017 at 7:51 am

Sir, it would be of an immense help to me if u can explain how to work with this module. i tried to open the homepage, but it's all Chinese. I am clueless. any help would be appreciated.

Reply

Jim Eli *says:*
July 24, 2017 at 4:44 pm

That's he purpose of this post. Please read it carefully.

Reply

Sukhbir Singh *says:*
July 25, 2017 at 4:56 am

is there a way to send AT commands over wifi from my laptop and is there a way to override the homepage of yunshan wifi relay

surya kant *says:*
August 6, 2017 at 4:32 am

hey! this tutorial was of immense help! it's awesome of you to share this knowledge with all of us beginners. but i have a question. Can i use this device to send live data via a udp channel or udp port so that i can monitor it live from a web server? Because I intend to use it to monitor the rpm of machines(bys giving input to optocoupler isolator pin G).

Reply

Lodie *says:*
August 28, 2017 at 11:57 am

Hi Jim

Thanks a lot, for this post, I managed to get the Yunshan IoT module up and running interfacing with Blink after Flushing it with your code… However now I am attempting to add additional Hardware, second Relay connected to GPIO14 and a second Opto coupler connected to GPIO12, I designated V10 to second Relay and V11 To second Opto coupler.
I duplicated the code where required, but seems to be stuck by my limited programing skills.
Please may I ask for some assistance.

Many Thanks
Lodie

Reply

Jim Eli *says:*
August 30, 2017 at 6:27 am

Peter *says:*
August 30, 2017 at 2:57 am

can you explain how to use the optical input

Reply

Jim Eli *says:*
August 30, 2017 at 6:19 am

See the TCP Client Demo program (GPIO5).

Reply

Lodie *says:*
September 4, 2017 at 12:08 am

Hi Jim

I actually constructed a totally new project on pc board , adding an ESP8266 module, The Relay and Opto coupler circuits are duplicated only IO Pin designation differs. With your version of the Sketch I am able to successfully operate GPIO4 (V0) and GPIO5 (V2). I would like to add GPIO14 (V3) As Output for the second Relay, and GPIO12 (V4) as Input from Opto Coupler.

Reply

DonDavio *says:*
September 30, 2017 at 10:16 am

Thank you so much for the board pinout and schematic diagrams. I bought a couple of these boards off AliExpress on impulse and they came with no documentation whatsoever (mine were marked “HW-622” for anyone else searching). Using your pinout and PlatformIO I got the relay up and clicking away on GPIO4!

Reply

przemek3122 *says:*
October 28, 2017 at 1:42 pm

Hi
It is possible to program this relay board using Arduino UNO microcontroller and its RX TX pins?

Reply

Jim Eli *says:*
October 28, 2017 at 3:03 pm

If it is possible to program the esp-8266 via an UNO, then yes. I have never tried this.

Reply

Kev *says:*
November 1, 2017 at 3:05 am

Hi Guys,

marked 'MZ' which means it has a reverse breakdown of 51 volts, but more importantly the spec sheet says it has a forward voltage drop of 3.5 – 5 volts. This means if you are trying to power it from 5 volts, it aint gonna work. Those of you using 12 volts will not notice a problem as the regulator will be getting around 7 volts anyway. 9 volt input could be marginal, especially driving the relay.
The simple way out is to short out the diode and make sure you get the power the right way around, or just parallel another normal rectifier diode across the existing one.

Cheers
Kev.

Reply

Allen *says:*
January 8, 2018 at 12:21 pm

This was a great article. I learned a lot. I modified your TCP Client Demo slightly and it works great. There are a couple of minor bugs in your code. I noticed that my DHCP server saw a different MAC than what your sample returned. The first thing I noticed is that the index to read the value should be 0 to 5 not 1 to 6. There are always 6 bytes so I don't know why you used the length function. Secondly you need to use WiFi.macAddress(mac); instead of WiFi.softAPmacAddress(mac); Here is my code:

WiFi.macAddress(mac);
String macID = String(mac[WL_MAC_ADDR_LENGTH – 6], HEX) + String(mac[WL_MAC_ADDR_LENGTH – 5], HEX)
+
String(mac[WL_MAC_ADDR_LENGTH – 4], HEX) + String(mac[WL_MAC_ADDR_LENGTH – 3], HEX) +
String(mac[WL_MAC_ADDR_LENGTH – 2], HEX) + String(mac[WL_MAC_ADDR_LENGTH – 1], HEX);

http:///open turns the relay on.
http:///close turns it off.
http:///MAC reports the MAC address.

Thanks again,
Allen

Reply

Dan Powell *says:*
March 20, 2018 at 10:48 am

Jim,
Thanks for the tutorial. Lots of good info. I have it now running. Curious, do you remember the settings you used on Arduino. I had the same problem with a 5V wall wort. Mine started whining after a couple minutes. Changed to 9V. I have it working on the Alexa. Works pretty well.

Again, thanks
Dan Powell

Reply

Jeff Silver *says:*
May 29, 2018 at 12:41 pm

Really useful page, thanks, as my module arrived without any documentation at all, not even polarity indication for power in. Can you explain what the "Optocoupler input" is? As far as I can tell from the example sketches, it's a simple digital input pin. I don't understand in what sense it is an "Optocoupler" input.

Ivan Vachovsky *says:*
May 30, 2018 at 10:41 am

Opto-coupler means that your sensor or whatever device you connect to this input will not connect electrically to the relay board. For example you can connect high voltage device there that otherwise will damage the board but with the opto-coupler it will not. Of course some precautionary measures have to be taken not to damage the opto-coupler itself. As far as 1wire sensors go, yes it is possible however proper additional programming needs to be done at ESP8266. It won't work as is.

Reply

Jim Eli *says:*
May 30, 2018 at 12:14 pm

Jeff,

>Can you explain what the "Optocoupler input" is?
An optocoupler is generally used to isolate separate circuits from each other, typically when different or dangerous voltage level exits between them. In the case of this device, the connectors on the board labelled F, G and H are separated from everything else. F is 5v output, G is the optocoupler input and H is an isolated ground. This is probably designed for incorporating a low voltage, human-operated switch, which would be isolated from high voltage main power.

>More specific question: can I use it as input for a DS18B20 one-wire temperature sensor?
I doubt it, the optocoupler is not bi-directional. I've never tried it.

Jim

Reply

Jeff Silver *says:*
May 31, 2018 at 2:08 am

Thanks Ivan, Jim. I understand the idea of optocoupler as an isolator; I just hadn't thought about the input side. That makes a lot of sense, and I agree (now that you've explained!) that it's not going to be bidirectional.
I can do one-wire programming, but the problem here is that none of the other IO pins are brought out to connectors. Any suggestions of a way to get access to one or more? My best (only!) idea so far is to solder a wire directly onto the ESP's pins; I do understand the risk to the chip of trying that!

Reply

Jim Eli *says:*
May 31, 2018 at 6:10 am

I wonder if you could solder to larger pin pad at the optocoupler? You could even remove the whole oc chip [https://www.youtube.com/watch?v=N_dvf45hN6Y]

Jeff Silver *says:*
June 1, 2018 at 1:05 pm

With a swift swipe of a very hot soldering iron, I managed to solder a lead directly onto the GPIO13 pad without either shorting anything out or overheating the chip. Thanks Jim for including the pinout on the board schematic. (I confirmed it with a voltmeter and a sketch that toggled it once a second.)
I now have a thermostatically controlled heater! Next step is to add a web interface for controlling it. (Back in the realm of software, where I'm much more comfortable!)
FYI, I'm powering it with a 6V/1A mains adapter.

June 1, 2018 at 4:39 pm

Fantastic!

Piyush Paliwal *says:*
August 5, 2018 at 6:45 am

Hey Jim, this article was immense help and I was able to work with this relay board. However there are some basic questions which am not getting and hoping you can help clear out the confusion.

– What are pins F,G & H used for? Although you have described the pins, I guess am lost there. Without connecting anything to these 3 pins, everything works normal for me.
– Pin F, you say it gives 5V out (correct me if I'm wrong here), can I use this current to power on the board i.e. on pin A?

Reply

Jim Eli *says:*
August 5, 2018 at 1:55 pm

Pins F, G and H would be useful for connecting some sort of switch/sensor that could control the relay. The 5V output would be used to power the sensor, it is not suitable for powering the board.

Reply

Piyush Paliwal *says:*
August 6, 2018 at 7:46 am

so, you are saying is that I can use these pins as a manual override as in I can control the light with manual toggle switch plus the ESP as well?
Also, do you have any practical use cases for these pins that you can think of out on top of your head?
Finally, all over the internet there is only this article related to this particular board, do you have to know of anything else which gives me basically the circuit diagram for this board.

Reply

Jim Eli *says:*
August 6, 2018 at 7:37 pm

1. Yes.
2. You just gave one yourself.
Circuit diagram is posted on my blog

Reply

Jim Eli *says:*
August 7, 2018 at 5:25 am

Simply wire a momentary switch between the 5V and optocoupler input lines. Both of my example programs demonstrate how to detect switch activation with simple debounce functionality. Good luck.

Reply

MH *says:*

H: Ground (isolated optocoupler input)
But i dont get any response when running your code example on my board. did i miss something?

thanks for your help

Reply

Jim Eli *says:*
September 19, 2018 at 5:12 am

You missed something. That's not how an optocoupler works: https://en.wikipedia.org/wiki/Opto-isolator

MH *says:*
September 19, 2018 at 9:29 am

i spend some time to understand how the optocoupler work and tried to find some examples how to connect everything…but now i am totally confused, because i don't find any example with three terminals.
it would be very kind, if you could give me a helping hand and show me how to connect a momentary button to switch the relay by pressing a button.
thanks for your support!

brian hambleton *says:*
August 27, 2018 at 8:40 pm

Thanks. This was the only place I could find a schematic for this board.

Reply

brian hambleton *says:*
August 28, 2018 at 6:03 pm

I bought mine from eBay in early '18. Out of the box, I couldn't get any response to the "AT commands". I re-flashed it with esp-easy.and it appears to be working fine. It behaves as all my other 8266 do. Hope this is of interest.

Reply

Wattana *says:*
September 25, 2018 at 4:30 am

Can this kit running uPython ?

Reply

Pingback: *Opensprinklette Single! | organic~monkey~motion*

Glen *says:*
November 24, 2018 at 5:53 am

Thanks for the post – As others have noted, information on this board seems to be scarce. Was about to throw mine in the bin before I found this page…

May still do so 😟 Am getting no response at all from the board via wifi, serial, or led. Have connected 10vdc supply, and traced power all the way through to U1 pin 2 – but am only getting 1.2vdc on L1, R4, C5 and C6. My interpretation is that the voltage regulator is faulty. My old eyes would have trouble with the replacement of U1, but my solder kit is nowhere near up

Thanks in anticipation

Reply

Dung_Nguyen *says:*
December 12, 2018 at 7:26 pm

Hello guys,

I bought 2 new board but I got problem during download coding. I connected ESP module to arduino mega 2560 as wiring below:
Arduino mega 2560 ESP module
TX ———- TX (J6)
RX ———- RX (J6)
GND ———- GND (J6)
9 VDC ———- (+) (J2)
GND ———- (-) (J2)
PC connected to arduino mega 2560 via comport COM4 then I try to donwload program (During download, The blue led on ESP8266 is flasing) but download failed and the message error show:
warning: espcomm_sync failed
error: espcomm_open failed
error: espcomm_upload_mem failed
error: espcomm_upload_mem failed
Can you please tell me where my mistake?
Note:
– Didn't remove J5 using for connected GPIO0 – GND during reboot and download
– Board selected on arduino is "Generic ESP8266 module"
Thank you for your help!

Reply

polargeek *says:*
December 12, 2018 at 9:10 pm

Pretty sure you have your RX and TX hooked up wrong it should be RX–>TX and TX–>RX

Reply

Dung_Nguyen *says:*
December 13, 2018 at 4:30 am

Even when changed as your recommendation the problem still have not been solved.
Additional when I connect RX–>TX and TX–>RX, the blue LED on ESP8266 chip didn't flashing?

Dung_Nguyen *says:*
December 13, 2018 at 4:39 am

Btw, When I made hooked up as above then open serial monitor window and boot the ESP module, some information reply look like connected has been established but It not respond with the AT command and I still cannot uploading my program
The message is "ets Jan 8 2013, rst cause :1, boot mode: (1,5)".

Antonio Solano *says:*
January 29, 2019 at 5:31 pm

Is it F: +5V output or input? Looking at schematic in U2 it states VIN for 5 V+. If I connect the Grounds: H to B, I am able to

Are my boards faulty? they don´t have the YunShan QR printed on the back !!!

Does someone face same problems configuring the INPUT in GPIO5?

```
// — Connected ouput pin.
#define RELAY_PIN 4 //ESP8266_GPIO4
// — Connected imput pin.
#define BUSY_PIN 5 //ESP8266_GPIO5
boolean busy;
// — Board Blue LED.
#define LED_PIN 2 //ESP8266_GPIO2
```

BR/A.S.

Reply

Jim Eli *says:*
January 29, 2019 at 5:35 pm

You can quote me, “F: 5V+ out”.

Reply

Antonio Solano *says:*
January 30, 2019 at 3:40 pm

Thanks Jim, but still same problem. If I shortcut the grounds B with H, and I try to give a pulse of 5V to G from F, the board reboots.

I also tried to use an external 5 volts charger, directly connected to G & H but nothing happens and when I measure the voltage beteween G and H it only messures 1 volt … any advice?

Reply

Jim Eli *says:*
January 30, 2019 at 5:56 pm

I have no idea what you are trying to accomplish. Are you trying to use the optocoupler as an additional input signal? The board is powered via A and B and needs about 7-9 volts to operate. The esp8266 can then control the relay which connects through C, D and E. F, G and H can be ignored unless you want to use them for an additional input signal.

Reply

Antonio Solano *says:*
January 31, 2019 at 4:30 am

Exactly, I want to use F, G and H for an additional input signal which will be disable the logic to operate the relay.

Reply

Antonio Solano *says:*
February 1, 2019 at 10:32 am

Dear Jim, I found the problem, the Diode was soldered in the wrong direction so the hardware was faulty. Thanks for your

Leo Zimmermann *says:*
February 14, 2019 at 11:06 am

Diode D5 soldered the wrong way around, check! Had the same issue. luckily I connected an open collector ciurcuit to the optocoupler input which made debugging a no brainer.

Thx very much @Jim for providing the schematic

Reply

µC eXperiment